

**Исследование параллельных методов программирования и совместного применения с соблюдением свойства изохронности в системе мониторинга воздушного пространства**

Н.И. Сурков<sup>1</sup>, Д.А. Пальгугев<sup>1</sup>, Ю.А. Сафонова<sup>2</sup>

<sup>1</sup> Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

603022, г. Н.Новгород, проспект Гагарина, 23.

E-mail: NikitaSurkov4x9@yandex.ru

<sup>2</sup> АО «Центральное конструкторское бюро аппаратостроения»

300034, г. Тула, ул. Демонстрации, 36

E-mail: yparshukova@yandex.ru

*В работе рассмотрены различные системы параллельного программирования и их гибриды для применения в системе мониторинга воздушного пространства и выполнения условия изохронности, т.е. требования проведения измерений и параллельных вычислений в распределенной системе в достаточно четко определяемые временные промежутки. Сделан вывод о необходимости применения комбинации нескольких методов и систем параллельного программирования для наиболее полной реализации условий изохронности.*

*Ключевые слова: изохронность, распределенные параллельные вычисления, параллельные методы программирования, системы мониторинга воздушного пространства.*

**Investigation of parallel programming methods and joint application with observance of the isochronic property in an airspace monitoring system**

N.I. Surkov<sup>1</sup>, D.A. Palguguev<sup>1</sup>, Yu.A. Safonova<sup>2</sup>

<sup>1</sup> Lobachevsky National Research Nizhny Novgorod State University.

<sup>2</sup> JSC "Central Design Bureau of Hardware Engineering".

*The paper considers various parallel programming systems and their hybrids for use in an airspace monitoring system and for meeting the isochrony condition, i.e., the requirements for measurements and parallel computing in a distributed system at fairly clearly defined time intervals. It is concluded that it is necessary to use a combination of several parallel programming methods and systems for the most complete implementation of isochrony conditions.*

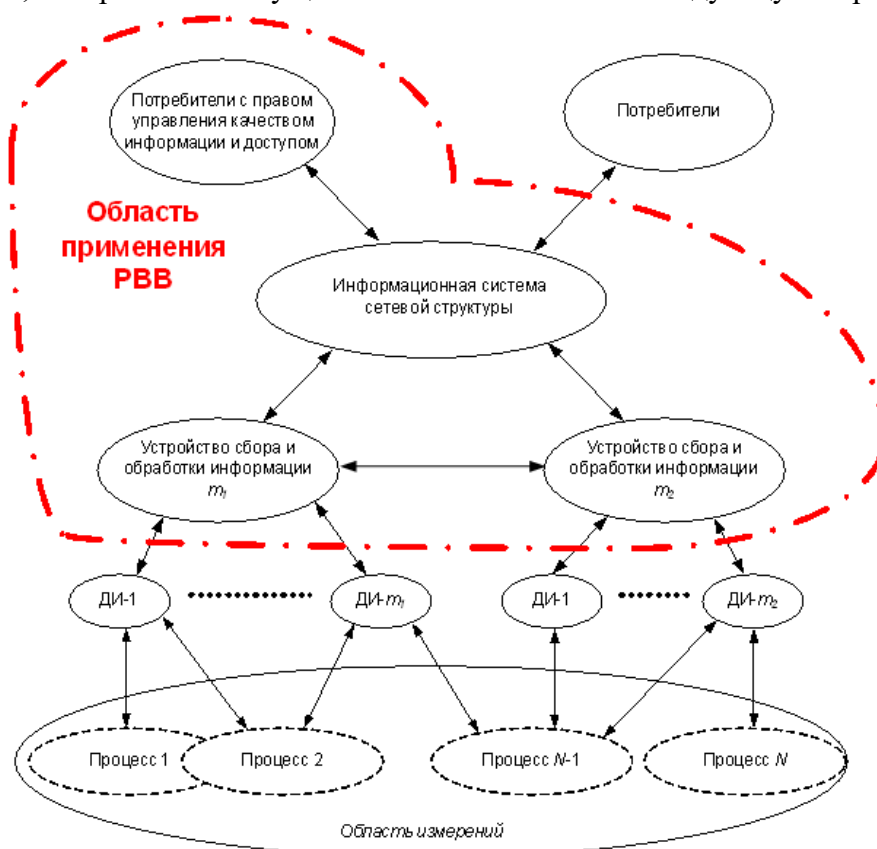
*Keywords: isochrony, distributed parallel computing, parallel programming methods, airspace monitoring systems.*

**Введение**

Системы мониторинга воздушного пространства предъявляют требования к изохронности, т.е. требования проведения измерений и параллельных вычислений в распределенной системе в достаточно четко определяемые временные промежутки. Если измерения, вычисления и последующая обработка информации проводятся без соблюдения этой особенности, происходит резкое уменьшение достоверности информации, она может быть значительно задержана во времени и внести существенные искажения при объединении её с информацией от других источников.

### Феноменологическая модель информационной системы сбора, обработки и обмена информацией сетевой структуры о прогнозируемых процессах дискретными датчиками-измерителями

Для анализа особенностей распределенных вычислительных процессов в информационных системах сетевой структуры, применяемых для сбора и обработки информации при мониторинге воздушной обстановки, воспользуемся феноменологической моделью информационной системы сбора, обработки и обмена информацией сетевой структуры о прогнозируемых процессах дискретными датчиками-измерителями (рис. 1, [1]). На рис. 1 в феноменологической модели выделена область применения распределенного векторного вычислителя (РВВ). Вместе с тем, при более детальном рассмотрении процессов измерения параметров воздушных объектов датчиками-измерителями (например, на рис. 1 ДИ-1 – Процесс 1, ДИ-1 – Процесс 2), не входящих в область применения распределенного векторного вычислителя, выявляются дополнительные требования к временным результатам процессов измерения датчиками-измерителями, которые имеют существенное влияние на последующую обработку.



**Рис. 1. Феноменологическая модель информационной системы сбора, обработки и обмена информацией сетевой структуры о прогнозируемых процессах дискретными датчиками-измерителями**

Так, например, траектория воздушного объекта (ВО) может быть представлена как сумма измерений во времени  $G_I(t)$  процесса 1 датчиком – измерителем 1 (ДИ-1):

$$G_1(t) = \sum_{n1=1}^{N_1} R_{n1} (t_{n1} \pm \Delta t_1), \quad (1)$$

где  $N_1$  – количество измерений датчиком ДИ-1 траектории ВО,  $R_{n1}$  – значения измеренных параметров (координат, составляющих скоростей),  $n1$  – порядковый номер

измерения 1-го процесса,  $t_{n1}$  – значения момента времени  $n1$ -го измерения,  $\Delta t_1$  – допустимые отклонения от времени измерения, связанные с производительностью ДИ-1, способом измерения датчика-измерителя, а также задержками при обработке в устройстве сбора и обработки информации (УСОИ) и в каналах передачи данных.

Данные по траектории ВО  $G_1(t)$ , как и данные по другим траекториям от ДИ-1, при поступлении в УСОИ распределенным векторным вычислителем не обрабатываются, т.к. получены от одного датчика-измерителя. Если процесс 1 одновременно измеряет ДИ-2, который пространственно разнесен и некогерентен ДИ-1, его измерения будут происходить в другие моменты времени  $t_{n2}$ , а траектория как сумма измерений ДИ-2 будет выглядеть следующим образом:

$$G_2(t) = \sum_{n2=1}^{N_2} R_{n2} (t_{n2} \pm \Delta t_2), \quad (2)$$

Измерения ДИ-1 и ДИ-2, если они следуют во времени друг за другом, обрабатываются РВВ на УСОИ, которое определяет идентичность измерений объекта (процесса 1), измеренного ДИ-1 и ДИ-2 в соответствии с методом и алгоритмом сетевой обработки [1].

Если к УСОИ  $m1$  подключено  $k$  – датчиков-измерителей, сумма измерений по всем датчикам-измерителям процесса 1, поступающая во времени на УСОИ, определяется следующим выражением:

$$G(t) = \sum_1^k G_k(t) = \sum_{n1=1}^{N_1} R_{n1} (t_{n1} \pm \Delta t_1) + \sum_{n2=1}^{N_2} R_{n2} (t_{n2} \pm \Delta t_2) + \dots \quad (3)$$

$$\dots + \sum_{nk=1}^{N_k} R_{nk} (t_{nk} \pm \Delta t_k),$$

Выражение (3) определяет совокупность последовательных измерений во времени  $k$ -датчиками-измерителями одного процесса – траектории воздушного объекта.

Изохронность измерений одного и того же процесса разными ДИ определяется соотношениями периода измерений (обзора РЛС)  $T_{изм\_k} = t_{(n+1)k} - t_{(n)k}$  и допустимым отклонением от времени измерения  $\Delta t_k$ , которое в общем случае разное для каждого ДИ, а кроме того, необходимо учитывать физический смысл измеряемого процесса и задержку передачи данных единичного измерения от ДИ до УСОИ. Если задержка превышает период измерения  $\Delta t_k > T_{изм\_k}$ , в этом случае изохронность процесса измерения датчиками-измерителями нарушается, текущее измерение вследствие задержки обработки будет недостоверным и не может быть принято в обработку на УСОИ, так как при этом не будет соблюдена очередность обработки измерений от датчиков-измерителей во времени.

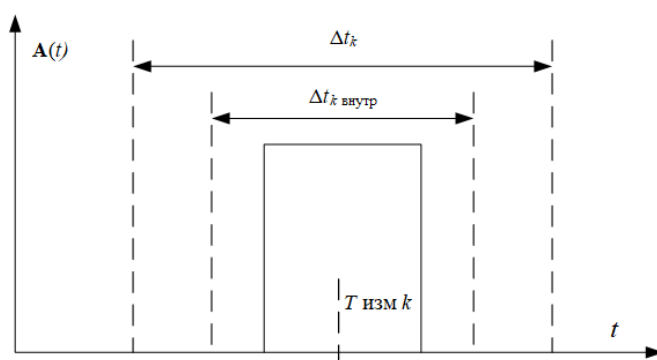


Рис. 2. Параметры изохронности процесса измерения

Параметры, определяющие требования к изохронности процесса измерения, показаны на рис. 2.

На этом рисунке видно, что отклонение длительности  $\Delta t_k$  включает отклонение длительности  $\Delta t_{k\_внутр}$ , обусловленное особенностями алгоритма обработки и производительностью датчика-измерителя.  $A(t)$  обозначен динамический массив данных датчиков-измерителей, в котором находятся результаты единичных измерений в конкретный момент времени.

Следовательно, условие изохронности распределенных процессов измерения и обработки в информационной системе в целом имеет следующий вид:

$$G(t) = \sum_{k=1}^k G_k(t) = \sum_{n1=1}^{N_1} R_{n1} (t_{n1} \pm \Delta t_1) + \sum_{n2=1}^{N_2} R_{n2} (t_{n2} \pm \Delta t_2) + \dots \quad (4)$$

$$\dots + \sum_{nk=1}^{N_k} R_{nk} (t_{nk} \pm \Delta t_k), \text{ при } \Delta t_{k\_внутр} \leq \Delta t_k \leq T_{изм\_k}.$$

Отметим, что  $\Delta t_k$  и  $\Delta t_{k\_внутр}$  могут принимать отрицательные значения в некоторых случаях, например, при переменном темпе обзора в РЛС с ФАР, и динамическим временным ресурсом обзора.

Результаты измерений отдельными датчиками-измерителями после обработки распределенным векторным вычислителем вносятся в динамический массив данных, обновляемый при последовательном поступлении соответствующих данных от датчиков-измерителей и не содержащих повторяющейся информации от разных датчиков-измерителей об одном и том же процессе.

Из выше упомянутого можно сделать вывод, что единичные результаты измерений процессов, которые не выполняют условие по изохронности измерений, не могут быть обработаны распределенным векторным вычислителем вследствие большой потери достоверности и невозможности экстраполяции на длительный временной интервал, а объединенные результаты не могут включать данные от датчика-измерителя, которые вследствие задержки во времени (по любым причинам) теряют свою достоверность. Из этого вывода следует, что при выполнении параллельных вычислений измерений протекающих во времени процессов необходимо учитывать время обработки в датчиках-измерителях и время доставки обработанной информации до устройств сбора и обработки информации по каналам передачи данных или по сети. Сформулированное условие является условием выполнения изохронности параллельных вычислительных процессов в большой вычислительной системе для случая наблюдения непрерывных, экстраполируемых процессов некогерентными дискретными датчиками-измерителями.

Ещё одним следствием из условия выполнения изохронности параллельных вычислений в большой распределенной системе мониторинга воздушного пространства является применение не только общепринятого вероятностного, но и временного подхода к объединению информации от нескольких источников. Именно стремлением «попасть» в изохронный интервал, обусловленный темпом обзора РЛС в многопозиционной радиолокационной системе, объясняется одна из задач работы [2].

### **Сравнительный анализ применимости методов параллельного программирования при проектировании программного обеспечения системы мониторинга воздушного пространства**

Сравнительный анализ применимости различных методов параллельного программирования проводится в контексте проектирования программного обеспечения систем мониторинга воздушного пространства.

В данной работе представлены только основные особенности из анализа ряда материалов, которые в сокращенном виде могут быть изложены следующим образом:

Linda [3]: отсутствие временной модели в спецификации, асинхронность, отсутствие готовых библиотек, малая распространенность, что приводит к необходимости реализации данного метода без основных базовых библиотек с обязательными дополнениями, высокая масштабируемость, слабая связность процессов.

OpenMP [4]: не поддерживает распределенную память, максимальное число потоков ограничено ресурсами одного вычислительного узла, как правило менее 256 потоков, стандартный компьютер 4 – 16 потоков, наличие готовых библиотек.

MPI [5]: полный контроль над изохронностью в распределенной системе, наличие готовых библиотек, широкая распространенность, гибкий контроль над процессом обмена данными между потоками. Один из минусов – сложность отладки и внедрения.

Для ограничения одновременного доступа разным УСОИ от пространственно-разнесенных некогерентных источников к обновлению данных об одном и том же процессе в динамическом массиве данных оптимальным представляется организация вычислений по системе параллельного программирования Linda [3] с некоторыми дополнениями, присущими особенностям изохронных параллельных вычислений. В системе Linda все параллельные процессы имеют доступ к общей памяти, единицей хранения данных которой является кортеж, а общее пространство памяти является пространством кортежей. Особенностью организации доступа к пространству кортежей является изъятие кортежа из пространства для внесения в него изменений. Процесс, который произвел изъятие, вносит изменения в кортеж и размещает его обратно в пространство кортежей. Изменить кортеж непосредственно в пространстве нельзя. Это правило исключает одновременный доступ и изменение единицы хранения. В динамической памяти кортежем выступает единица хранения – столбец обработанных данных об объекте за последний текущий момент времени. Для выполнения требования изохронности можно установить минимальное время, в течение которого любой из параллельных процессов не может вносить изменения в конкретный кортеж. Если в течение этого времени запрета изменения измерения по конкретному объекту стали неактуальны или появились более новые в другом параллельном процессе, то доступ получит процесс с более новыми измерениями. Время последнего измерения вносится в кортеж при его возвращении в пространство кортежей.

Необходимо отметить определенное удобство системы Linda, которое заключается в том, что в любой последовательный язык достаточно добавить четыре функции, чтобы обеспечить параллельное программирование. Описание функций достаточно подробно представлено в [3].

Наиболее подходящими в рассматриваемой предметной области мониторинга воздушного пространства являются системы OpenMP и MPI.

### **Гибридный подход параллельного программирования в системе мониторинга воздушного пространства**

Основные положения по созданию комбинированной системы параллельного программирования, предназначенной для решения задач в системе мониторинга воздушного пространства, по мнению авторов, можно кратко изложить следующим образом.

В основу комбинированной системы предлагается положить модифицированный метод Linda. Узлы обработки информации (размещенные на УСОИ) периодически генерируют кортежи с собранной информацией и помещают в распределенное пространство кортежей, другие узлы забирают их для обработки. Модификация заключается в добавлении специальных кортежей-маркеров, которые запускают

обработку или сигнализируют о наступлении определенного временного интервала, а также добавление точных временных меток к каждому собранному кортежу, чтобы процессы обработки могли фильтровать кортежи по временным меткам (в пределах требуемого временного окна), таким образом достигается изохронность параллельных процессов.

OpenMP не предназначен для распределенной по системе памяти, но его можно использовать вместе с другими методами [6], в данном случае с Linda. Применение заключается в распараллеливании процессов внутри каждого узла по отдельности, а не всей сетевой структуры вместе (рис.3).



**Рис. 3. Схема комбинированного метода параллельного программирования**

Благодаря свойству инкрементальной параллелизации появляется возможность постепенно внедрять метод в узлы без глобального изменения кода, распределяя сбор и обработку данных по потокам. Изохронность достигается путем планирования на основе времени и синхронизации потоков с помощью барьеров.

Подход, основанный на разбиение системы на узлы и их распределении как по процессам, так и по потокам, показал эффективность в исследовании [7]. Работа посвящена разработке и исследованию реализации версии модели мелкой воды, в которой MPI используется между узлами, а OpenMP — внутри каждого узла. На небольшом числе узлов чистый MPI подход на основе обмена сообщениями оказывается эффективнее гибридного. Это связано с накладными расходами на создание и синхронизацию потоков, а также с тем, что при малом числе узлов нагрузка на коммуникационную сеть мала, и выигрыш от снижения числа межузловых пересылок не компенсирует издержки многопоточности. Но по мере увеличения числа узлов, начиная с шести и более, гибридный подход начинает превосходить чистый обмен сообщениями. Сокращение объема межузловой коммуникации за счёт обмен через общую память внутри узла показывает прирост производительности.

В дальнейших рассуждениях, вместо MPI предлагается координация через модифицированное пространство кортежей Linda, что является одним из основных условий обеспечения свойства изохронности.

## **Выводы**

Проведено исследование возможности использования методов параллельного программирования при создании системы мониторинга воздушного пространства, где имеется требование проведения измерений и параллельных вычислений в распределенной системе в достаточно четко определяемые временные промежутки. Обнаружено, что классическая реализация модели Linda не подходит под условия системы, в следствии чего необходимо применять модификации и дополнительные методы параллельного программирования.

Предложен гибридный метод параллельного программирования, в основе которого используются процессы Linda с изменениями в части временных ограничений по внесению изменений в пространство кортежей для соблюдения требований изохронности обработки информации от датчиков-измерителей, а также расположение потоков OpenMP на узлах обработки данных. Такой подход позволяет связать программное обеспечение в единую изохронную систему, начиная с отдельных узлов и продолжая на сетевом уровне.

## **Литература**

1. Пальгуев Д.А. Система сбора, обработки и управления качеством информации сетевой структуры от пространственно разнесенных дискретных датчиков-измерителей о прогнозируемых процессах // Проектирование и технология электронных средств № 4. 2022. с. 14-20.
2. Пальгуев Д.А. Особенности вторичной и третичной обработки информации о воздушных объектах в многопозиционных пространственно-разнесённых некогерентных радиолокационных системах / Пальгуев Д.А., Ильясафов А.Д., Пиунов К.Н., Багурин А.С. // Радиотехнические и телекоммуникационные системы. 2024. № 2. с. 52-58
3. Воеводин В.В., Воеводин Вл. В. Параллельные вычисления: учебное пособие. СПб.: БХВ-Петербург. 2002. 608 с.
4. Соснин В.В., Балакшин П.В., Шилко Д.С., Пушкарев Д.А., Мишенёв А.В., Кустарев П.В., Тропченко А.А. Введение в параллельные вычисления: учебно-методическое пособие. СПб: Университет ИТМО. 2023. 128 с.
5. Волосова А.В. Параллельные методы и алгоритмы: учебное пособие. М.: МАДИ. 2020. 176 с.
6. Антонов А.С. Параллельное программирование с использованием технологии OpenMP: учебное пособие. М.: МГУ. 2009. 77 с.
7. Чаплыгин А.В. Гибридная модель мелкой воды с использованием технологий MPI-OpenMP / Чаплыгин А.В., Гусев А.В. // Проблемы информатики. 2021. №1 (50). С. 65-82.
8. Gropp, W. Using MPI: Portable Parallel Programming with the Message-Passing Interface/ W. Gropp, E. Lusk, A. Skjellum. — 3rd ed. — Cambridge. Mass.: MIT Press. 2014. 310 p.